

面向SOA的E/E架构设计数智化转型实践

金劭南 高仕宁 李超 吴振举 陈泓宇 柳菁

(中国第一汽车股份有限公司研发总院, 长春 130013)

【欢迎引用】金劭南, 高仕宁, 李超, 等. 面向SOA的E/E架构设计数智化转型实践[J]. 汽车文摘, 2024(3): 31-38.

【Cite this paper】JIN S N, GAO S N, LI C, et al. Digitalization & Intelligence Transformation Practice of SOA E/E Architecture Design [J]. Automotive Digest (Chinese), 2024(3): 31-38.

【摘要】随着数智化潮流的推进,使用工具链进行整车架构开发已成为传统车企数智化转型的关键。通过文献研究、实操探索等方式,结合集中式面向服务的架构(SOA)项目开发实例,研究使用EA、PREEvision工具链进行架构设计数智化转型实践,阐述相关工具在整车功能架构、软件架构设计中的应用。同时根据工具在应用过程中存在的问题,进行二次插件开发,解决架构设计过程中存在的瓶颈。通过案例实践表明,使用上述两款工具可有效应对整车E/E架构设计数智化转型,提高研发效率。

关键词: 整车架构 数智化转型 工具链 架构设计 功能设计

中图分类号: U463.61 文献标志码: A DOI: 10.19822/j.cnki.1671-6329.20220288

Digitalization & Intelligence Transformation Practice of SOA E/E Architecture Design

Jin Shaonan, Gao Shining, Li Chao, Wu Zhenju, Chen Hongyu, Liu Jing

(Global R&D Center, China FAW Corporation Limited, Changchun 130013)

【Abstract】As the trend of digital intelligence advances, using tool chain to develop vehicle architecture has become the key to the intelligent transformation of traditional automobile enterprises, through literature research, practical exploration and other ways, develop examples in conjunction with centralized SOA architecture projects, This paper studies the use of EA and PREEvision tool chain to carry out intelligent transformation practice of architecture design, describes the application of relevant tools in the design of vehicle functional architecture and software architecture. At the same time, according to the problems existing in the application process of the tool, the secondary plug-in development is carried out to solve the bottleneck in the architecture design process. The case practice shows that the above two tools can effectively cope with the intelligent transformation of vehicle E/E architecture design and improve the research and development efficiency.

Key words: Complete Vehicles Architecture, Digitalization & Intelligence Transformation, Tool Chain, Architecture Design, Function Design

0 引言

随着汽车“新四化”(智能化、网联化、电动化和共享化)的快速发展,汽车上的功能需求越来越丰富,车辆ECU控制器数量大幅度增长,汽车电子通信网络越来越复杂,对整车E/E架构的要求也越来越高^[1]。传统分布式汽车电子电气架构存在算力不足、布线复杂和高耦合性的缺陷,已然不能够支撑当前的技术挑战及市场需求。因此在软件定义汽车的背景下,各大主机厂纷纷提出了面向服务的架构(Service-Oriented Architecture, SOA)。如Volvo-SPA2.0架构,其设计主要

原则为标准化服务,服务松耦合,构建可重复利用的服务来实现粗粒度、松耦合、灵活性强的集中式汽车E/E架构^[2]。

数字化时代,传统企业“数智化转型”已成为提升效率的必由之路^[3]。企业数智化转型是利用现代信息技术对企业业务进行重塑,运用主流、前沿的智能化技术对业务流程及机制进行优化和管理,实现业务效率的提升和创新,保持企业的竞争力。在软件定义汽车的时代背景下,数智化转型已在汽车行业内逐步应用,成为重塑产业结构的重要动力。很多主机厂都提出了数智化转型之道。丰田作为数智化转型的标杆,

独创“一页纸极简思考法”^[4];大众也提出了相应的数智化转型计划,成立汽车软件组织 Car.Software,力争2030年成为软件驱动型的移动出行服务提供商。

智能网联汽车数智化,学习互联网思维,实现产品敏捷开发,即数字汽车开发^[5]。针对面向SOA的电子电气(Electrical&Electronic, E/E)架构设计数智化转型,各传统车企均处于探索实践的阶段,在这其中使用合适有效的工具链可以有效提升研发效率^[6]。本文在SOA架构思维下,探究整车E/E架构设计中功能架构设计和软件架构设计数智化转型的解决方案并实践,同时根据工具链间存在的痛点问题,对工具链进行二次开发,满足业务灵活度及契合度。

1 面向SOA的E/E架构设计

1.1 传统E/E架构设计

以传统基于域控制架构开发的实际流程为例,基于域控制器的传统分布式架构开发是以车型装备定义作为输入,在各阶段输出相应的功能设计规范传递给下游,指导其系统开发,传统E/E架构设计如图1所示。

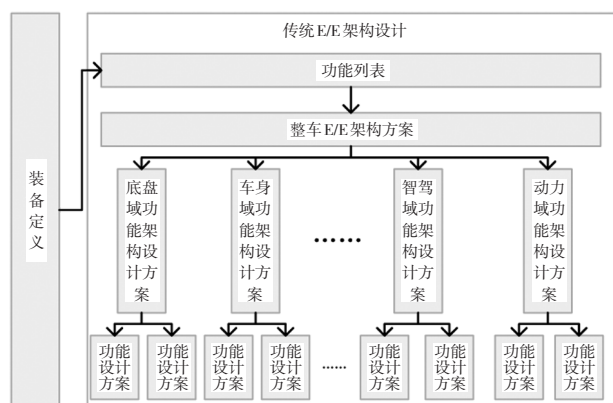


图1 传统E/E架构设计

基于域控制架构开发的核心是以控制器为主线进行的架构设计。首先通过装备定义明确车型定位,在架构设计阶段,依据装备定义明确车型需要的所有功能,输出功能特性列表。再对功能进行功能域划分,结合ECU控制器按照功能域输出架构设计方案。在此基础上输出每个功能的详细功能描述方案。然而在“软件定义汽车”的时代下,功能需求呈爆发式增长,该设计模式不能够完美适应新架构思想下的开发设计,且传统开发流程中各环节的输出物为文档,并未合理使用相关工具链,难以顺应当前传统车企数智化发展趋势^[7]。

1.2 面向SOA的E/E功能架构设计

相较于传统的E/E架构开发,面向SOA的架构开

发更侧重于降低软件的耦合度,抽象并暴露出更多的服务。本文参考AUTOSAR^[8]标准中关于方法论的描述,以集中式SOA架构项目为例,提出一种基于集中式SOA架构的功能开发方案,如图2所示。

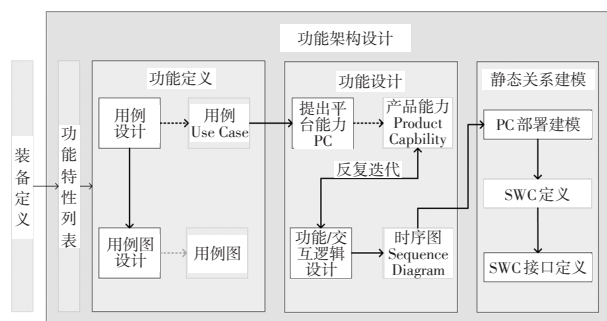


图2 集中式SOA功能架构设计

基于图2的功能架构设计,SOA的功能架构开发过程中,同样以装备定义作为输入,用于描述整车产品的定位。不同于传统架构开发以电子控制单元(Electronic Control Unit, ECU)控制器作为主线,SOA的功能架构开发过程中更侧重于功能和服务^[9]。输出功能特性列表后进行功能定义、功能设计和服务抽象。借鉴互联网开发的思维,在此过程中输出用例图和时序图给下游专业。其中用例图用来描述功能的定义内容,功能设计则是对车辆平台软件和硬件的能力分析,提取出产品能力(Product Capability, PC),并描述各能力之间的交互,最终形成时序图。

1.3 面向SOA的E/E软件架构设计

集中式SOA架构定义了整车级SOA软件架构,通过分层部署,提取出全局变量的服务(如车速、时间、车辆状态等),并将其作为通用的软件模块服务,以实现基础功能软件接口统一,并可灵活部署(图3)。

上述工作需要在项目开发一开始便设计好,之后的软件架构工作均需围绕其开展。结合集中式SOA架构项目下整车E/E架构开发流程,其软件架构设计方案如图4所示。

在软件架构设计之初,便需要对整车SOA架构进行合理分层以及模块划分,搭建整个软件架构的框架,用于承接功能架构所输出的平台能力(PC)。

当接收到功能架构设计所输出的平台能力(PC)后,由软件架构进行承接,对PC进行模块部署,落到对应的模块中,之后需要对模块进行设计(Module Design)。软件架构设计即根据平台能力PC在Module上进行软件组件(Software Component, SWC)划分,并以输入的平台能力(PC)为参考,对SWC进行接口设计。将上一节内容产出的PC对功能的描述转化为软

件描述。

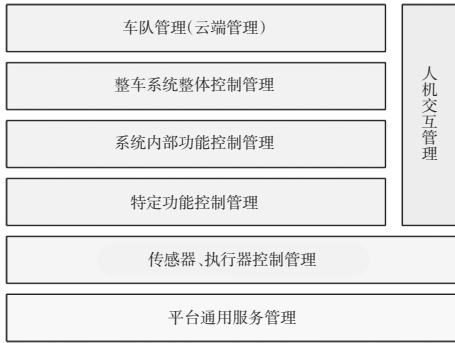


图3 软件模块分层部署示意

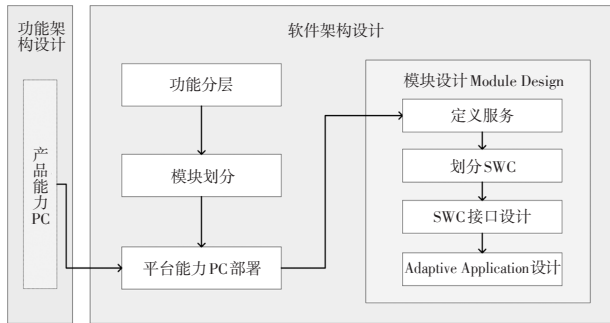


图4 面向SOA的软件架构开发流程

2 面向SOA的E/E架构数智化转型工具链

在数智化大背景下,使用合理的工具链进行开发设计可有效地提升研发效率^[10-11],高效对产出物进行管理,实现多人协同办公。结合集中式SOA架构项目实际开发过程,阐述相应工具链的使用情况。

2.1 软件生命周期设计工具——Enterprise Architect

Enterprise Architect (EA)是 SparxSystems 公司开发的旗舰产品,是一款基于统一建模语言(Unified Modeling Language, UML)的可视化建模与设计工具^[12-13]。Enterprise Architect是一个对于软件系统开发有着极好支持的计算机辅助工程软件(Computer Aided Software Engineering, CASE)。不同于VISIO等其他UML建模工具,CASE能够支撑系统开发全过程,且支持多人协同操作,满足团队开发设计,同时能根据项目实际需求进行二次开发,有效提高建模设计效率及管理能力(图5)。

结合面向SOA架构开发过程中功能架构开发流程的相关活动,EA主要用于功能定义设计及模块、SWC定义等相关环节。主要建模内容如下:

(1)用例(Use Case, UC):系统功能基本描述,结合项目实际情况。

(2)用例图:描述参与者与系统功能间的交互形式。

(3)状态机图:描述了一个对象在其生命周期内所经历的各种状态,以及状态之间的转移、发生转移的原因、条件和转移中所执行的活动。

(4)产品能力PC:描述整车产品所提供的车辆能力。

(5)时序图:描述平台能力PC之间发送消息的时间顺序显示多个对象之间的动态协作。

(6)SWC: AUTOSAR 标准制定目标之一,将PC的功能描述转化为软件描述。

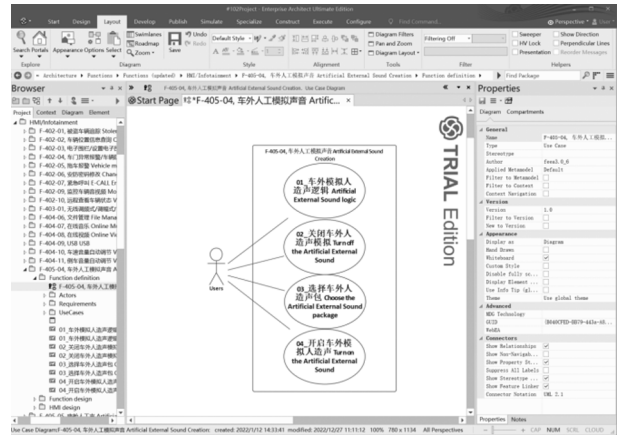


图5 EA建模工具

2.2 软件架构设计工具——PREEvision

PREEvision 提供给开发人员完整的协同开发平台,可用于基于AUTOSAR的软件架构设计,支持SWC的创建划分以及接口的定义设计,为软件架构的定义和端口连接提供图形支持,同时PREEvision还支持不同版本AUTOSAR软件技术规范的导入和导出^[14-15]。

使用PREEvision进行软件建模的优势主要有2点:

(1)PREEvision支持AUTOSAR标准,满足集中式SOA架构项目的建设需求。

(2)PREEvision能够很好的进行全链条的追溯,当其中某一环节发生改动时,PREEvision能够展示出所有与之关联的变动,供开发人员进行参考。

3 面向SOA的E/E架构数智化转型实践

在集中式SOA架构项目开发过程中尝试使用EA、PREEvision相关工具链进行架构模型的构建。其中,使用EA实现功能架构设计相关内容,包括上述多种UML绘制、SWC和接口定义;使用PREEvision实现软件架构设计内容,包括服务定义、SWC、接口的定义和设计,以及自适应应用(Adaptive Application)设计。本章以信息娱乐域中“车外模拟人工声音”功能

为例,阐述当前面向SOA的E/E架构开发数智化转型实践情况。

3.1 基于EA的电子电气功能架构设计

在功能架构开发阶段,主要使用EA进行UML模型的绘制。完成功能定义、功能设计及平台能力PC部署建模。

3.1.1 功能定义

功能定义阶段主要是将用户需求转化为工程需求,并运用行业标准UML语言进行建模规范。在功能定义阶段主要工作是使用EA来定义功能用例(UC),并绘制UC图。在设计每个功能用例时,要明确以下属性:

(1)描述:包括UC名称及UC的详细描述,对本UC进行简单描述。

(2)前置条件/后置条件:描述该UC生效的前提条件和触发UC后的状态。

(3)场景:描述UC在触发期间整个事件流,包括基本事件、异常事件和备选事件。

(4)非功能需求:为满足功能而必须具有的功能需求以外的特性,包括安全性、可靠性、法规要求和时效性。

本文以“车外模拟人工声音”为例将整个大功能的使用场景通过需求分析拆分成多个UC。

当定义好UC后,还需定义“参与者”,即描述触发用例的角色,如驾驶员和乘客。还需要绘制时序图,建立“参与者”、“用例”和非功能需求之间的联系,如图6所示。

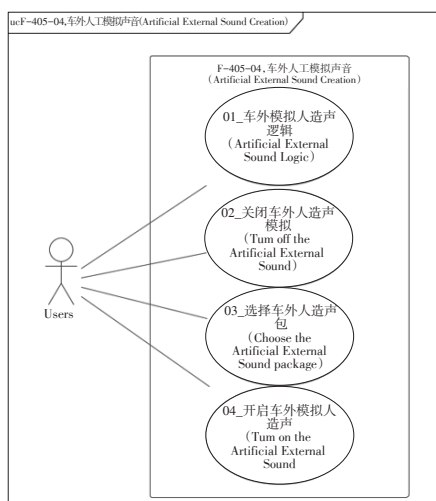


图6 车外模拟人工声音用例

在功能设计阶段使用EA工具可以很好地使用工程描述语言表示出用户与功能之间地交互关系,供相关专业开发人员进行开发。

3.1.2 功能设计

完成用例设计后,需要对每个用例进行功能设计,即为实现功能所需要的时序设计。该阶段同样在EA工具上使用UML建模语言进行动态交互图设计。将UC涉及到的信息,功能实现过程和场景进行细化,参考功能定义中的前置条件、后置条件、功能场景进行设计,在设计过程中可以不断发现功能设计阶段所存在的遗漏点和不合理之处,不断优化迭代。

功能设计首先根据功能定义中的信息提出平台能力(PC),明确为实现全部功能整车需要提供什么样的能力,以“车外模拟人工声音”为例,为实现这个功能需要车辆具备模拟发动机声音的能力,同时根据不同车速发出不同音效,还需要车辆具备获取车速的能力。在设计PC时,需要考虑如下因素:

(1)PC名称及描述:描述PC所提供的能力

(2)PC相关操作:描述PC之间交互的操作,如获取车速信息,获取车辆状态;

(3)PC相关操作返回数据类型:PC间交互后返回数据的类型,为基本数据类型或结构体。

本文以“车外模拟人工声音”功能为例,给出EA工具链PC设计过程。PC设计过程见表1及图7所示。

表1 车外模拟人工声音PC设计(部分设计)

PC名称	PC描述	PC操作	类型
VI_HUML_ProvideLogicalArtificialExternalSoundCreation	提供车外人造声的主控制。	LogicalExternalSoundCreation_enum_RequestStatusofArtificialExternalSoundCreation	enum
Chassis_SA_Brake_ProvideVehicleSpeedInformation	提供车速信息	VehicleSpeed_double_GetVehicleSpeed	double
HMI_View_ProvideArtificialExternalSoundCreation	提供车外人造声的代理显示输出		
PT_VCoord_PTCO_ProvideTorqueCoordination	提供总成扭矩值及状态	PTCO_SubscribeShiftLvrPosnIC_PT	enum

在初步提出PC后,需要使用这些提出的PC进行功能时序设计,其目的是通过同台交互设计使用这些PC,共同协同工作实现整个用例UC。该过程使用UML时序图进行工程描述。在绘制时序图时,主要有考虑到如下因素。

(1)对象:实现功能的主要对象,即消息实体。在此主要包括参与者与上一步骤中所提到的PC。

(2)生命线:即功能实现的参与生命线。

(3)消息:各对象间交互的内容,实现功能事件的

动态交互,主要用到了同步消息与异步消息,依据实现的内容不同来进行选择。

(4)组合片段:包括一些条件判断框,如选项片段(Opt)、备选条件进入框(Alt)、并行执行框(Par)等。用于处理用例(UC)中场景信息涉及到的不同情况。

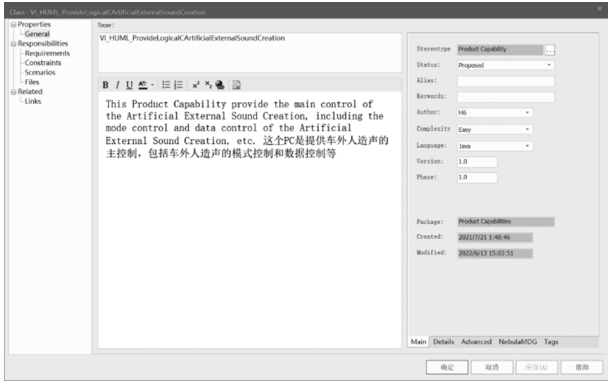


图7 EA上进行PC设计

本文以“车外模拟人工声音”功能中的功能设计为例,给出功能设计示例,如图8所示。

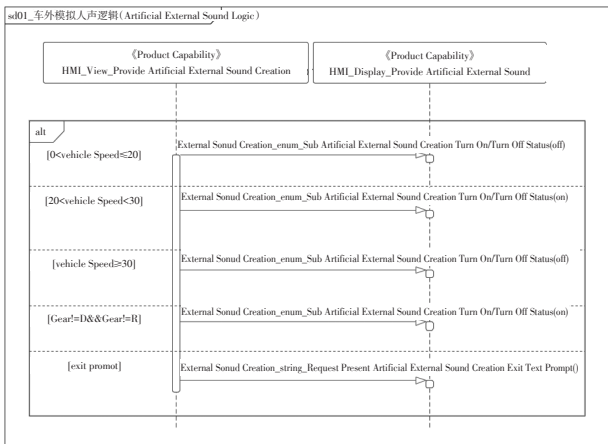


图8 用例-车外模拟人工声音功能设计示例

在绘制时序图过程中可以检验上一步设计PC的合理性,保证每个PC的颗粒度是合理的,能够覆盖整个功能的完整生命周期,因此在功能设计阶段PC的设计以及时序图设计是一个相互迭代的过程,保证最终功能设计结果是合理的。

3.1.3 SWC及其接口定义

为了能够建立PC上下游间的追溯关系,功能架构设计阶段还需要在EA中完成PC的部署录入工作,建立SWC及其接口,需要说明的是在EA中并不进行Module及SWC的设计工作,仅完成定义工作。其目的有2点:(1)方便对PC进行管理,后续PC可以对其他功能进行复用,方便PC的快速查找、迭代更新及管理。(2)在EA进行建模后可以提供良好的全链条关系追溯能力,后续可对每个功能的UC—PC—Module—

SWC—接口进行全链条追溯,提升开发效率(图9)。

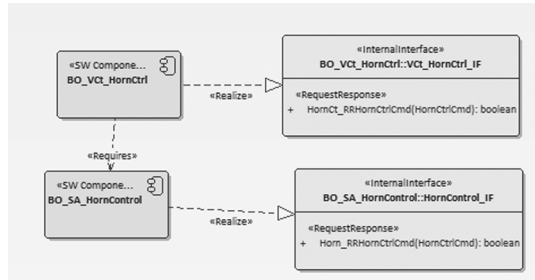


图9 EA工具上SWC及接口设计

3.2 基于PREvision的电子电气软件架构设计

3.2.1 基于PREvision的分层与模块设计

在进行软件架构开发之初便需制定好整个E/E架构分层设计与模块设计,通常情况下该结构后期不会更改,用于支撑整个功能架构及软件架构的开发,依据1.3节中图3的划分思想,本文按照功能域给出服务接口层级设计,如图10所示。

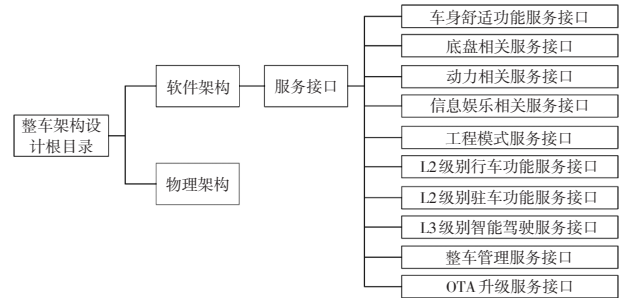


图10 服务接口层级设计

本项目依照图10的服务层级设计对整车进行接口层级划分,保证后期的维护和管理,并在PREvision里进行结构创建,如图11所示。

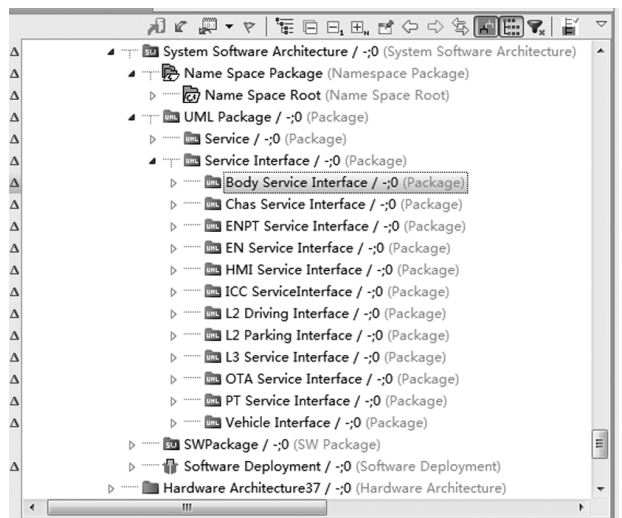


图11 PREvision分层结构设计

在PREvision中创建好结构后,后续的服务及SWC建模均在此结构上进行,因此在架构设计之初分

层与模块设计非常重要,决定后续软件架构开发的合理性。

3.2.2 基于PREvision的服务设计

在接收到功能架构的PC输入后,在软件架构开发阶段需要先进行服务抽象,将PC转化为服务。在AUTOSAR标准的指导下^[8],PREvision支持开发二次插件脚本实现自动建模,导入标准格式的Excel并自动完成服务功能的搭建。为了保证准确导入,在服务设计阶段,需要按照规范格式编写软件服务Excel表格,二次开发脚本对输入表格进行解析,执行插件脚本即可导入服务模型到PREvision,以车外模拟人工声音功能为例,设计车速相关的功能服务,其解析前的Excel如表2所示。

表2 车外模拟人工声音服务设计表(车速相关服务)

服务接口元素名称	服务接口元素类型	方法参数名称
Brk_NotifyCycleWheelSpeedRR	Event	Brk_NotifyCycleWheelSpeedRR
Brk_NotifyCycleWheelSpeedFL	Event	Brk_NotifyCycleWheelSpeedFL
Brk_NotifyCycleWheelSpeedFR	Event	Brk_NotifyCycleWheelSpeedFR
Brk_NotifyCycleWheelSpeedRL	Event	Brk_NotifyCycleWheelSpeedRL
Brk_NotifyCycleVehicleSpeed	Event	Brk_NotifyCycleVehicleSpeed

PREvision导入上述excel表并进行解析,在工具中自动创建相关服务接口,创建结果如图12所示。

```

▶ ---□ Brk_NotifyCycleWheelSpeedFL_E:Brk_NotifyCycleWheelSpeedFL_E (SW Port)
▶ ---□ Brk_NotifyCycleWheelSpeedFL:Brk_NotifyCycleWheelSpeedFL (SW Port)
▶ ---□ Brk_NotifyCycleWheelSpeedFR_E:Brk_NotifyCycleWheelSpeedFR_E (SW Port)
▶ ---□ Brk_NotifyCycleWheelSpeedFR:Brk_NotifyCycleWheelSpeedFR (SW Port)
▶ ---□ Brk_NotifyCycleWheelSpeedRL_E:Brk_NotifyCycleWheelSpeedRL_E (SW Port)
▶ ---□ Brk_NotifyCycleWheelSpeedRL:Brk_NotifyCycleWheelSpeedRL (SW Port)
▶ ---□ Brk_NotifyCycleWheelSpeedRR_E:Brk_NotifyCycleWheelSpeedRR_E (SW Port)
▶ ---□ Brk_NotifyCycleWheelSpeedRR:Brk_NotifyCycleWheelSpeedRR (SW Port)

```

图12 PREvision脚本生成服务(车速相关服务)

3.2.3 基于PREvision的SWC设计

在AUTOSAR中,软件架构由SWC组成,通过服务接口的软件实现。软件架构开发过程中依据传递过来的PC进行SWC设计,并在PREvision中进行实现。本文以AP建模规范为例,依据所实现的服务来创建SWC,在Root Composition中新建用于AP的组成(Composition),并在该Composition中新建对应的自适应应用软件组件(Adaptive Application SW Component),如图13所示。

3.2.4 基于PREvision的SWC接口设计

定义好SWC后,还需添加对应的Adaptive Re-

quired/Provided Port,建立服务于SWC之间的关系,在此步骤中需要重新命名并将Required Port和Provided Port建立连接。将需要进行AP设计的Service Interface直接拖拽至AP Required/Provided接口。以获取车速信息的SWC为例,在PREvision中SWC接口设计如图14所示。

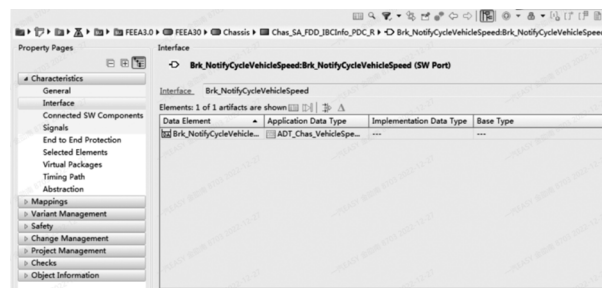


图13 PREvision中创建SWC(获取车速信息)

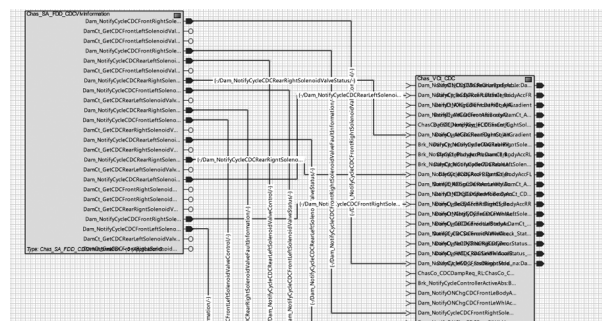


图14 基于PREvision的SWC设计

3.2.5 基于PREvision的Adaptive Application设计

在AUTOSAR方法论中,AP是可执行文件(Executables)的集合,一个Executable源于一个AP SWC,在PREvision中设计方法如下。

(1)在System Software Architecture下新建SW Package,在此Package下新建两个名称为Application和Executable的SW Package,在两个SW Package下分别创建Adaptive Application以及Executable。

(2)在Adaptive Application的General属性窗口,选择Adaptive Application Kind为Application Level或者Platform Level。

(3)在Adaptive Application栏Adaptive Application Editor中将APP与Executable关联,并填写Adaptive Application Version信息,Executable Build Type及Version信息,将AP SWC与对应Adaptive Application建立关联。

Adaptive Application设计完成后,最终以Application为单位,导出ARXML供下游专业进行软件开发。

3.3 关键工具链二次开发设计

使用工具链进行面向SOA的E/E架构开发可以

提升研发效率,提升了模型管理及维护能力,减少容错率,一定程度上保证了模型的严谨性。但工具并非万能,一方面开发过程中使用工具可有效提升工作效率,实现一定程度上的数智化转型,另一方面工具的局限性可能会带来一些重复冗余的工作,给开发设计增添了工作量。无论是EA还是PREvision,都并非能按照实际开发需求进行完美适配,在开发过程中难免会存在一定的局限。本文结合项目实践过程中发现的问题进行总结并给出解决方案:

(1) 对建模内容管理困难

在开发过程中,EA上维护了超过300个功能、3000个UC、6000个平台能力PC。EA工具缺乏对这些功能的导出,只能在工具上进行查看,对于不使用EA的开发人员来说,不方便查看建模内容。

(2) 无法高效产出功能需求文档

在面向SOA架构开发需要产出功能需求文档(Product Requirement Document, PRD)传递给下游专业进行实现,目前在EA上实现的建模内容只能通过手动粘贴的方式来编写PRD,在此过程中产生了一些不必要的工作,且容易出现人为失误。

(3) EA工具缺乏对全链条追溯的能力

UC、PC等模型元素虽然可通过建模的方式进行关联追溯,但EA工具链中缺少相关功能能够直观展现这些关系,且不支持全链条关系导出供开发人员参考,在实际工作中会产生一定的麻烦。

(4) EA工具与PREvision工具间未实现打通

从前两节的实践操作中可以看出,在架构开发中主要使用的工具链为EA和PREvision,2个工具相互独立,且在各自的工作范围内有重复的工作,增添了冗余的操作,一定程度上降低了工作效率,且不能实现联动更新,为后期维护增添了困难。

针对上述存在的问题,考虑到EA以及PREvision均可支持二次插件开发来提高生产效率,如图15所示。

本文结合工作实际情况下存在的问题瓶颈,在插件开发过程中给出解决问题的思路及办法。

(1) 针对EA上模型管理困难的问题,可以考虑通过EA二次开发插件程序,抓取UC、PC元素并获取到模型元素中的信息,通过Excel表的形式导出,供架构开发人员进行管理。考虑到功能数量繁多、模型信息需要人为录入缺乏规范性管理,可以制定相应的规范标准,制定属性信息描述范式与命名规则。同时可以按功能域进行划分,对用例进行管理。



图15 EA二次开发插件

(2) 针对EA模型无法导出功能需求文档PRD的问题,同样可以使用二次插件开发的方法进行解决。首先需要规定好PRD文档的模板,通过抓取EA中用例图、时序图及相关模型元素的信息,导出PRD初版文档,该文档中包含在EA中建模所有内容,PRD中其它与模型无关的(如用户故事、法规项等)内容在导出后的文档中补充即可,避免重复操作,减少开发人员的工作量(图16)。

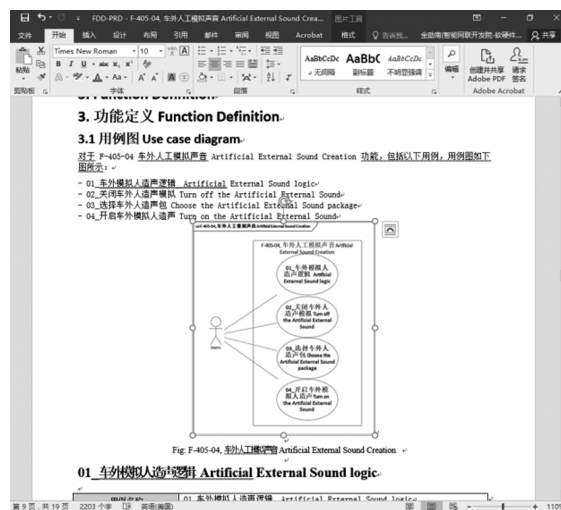


图16 EA二次开发一导出PRD文档示例

(3) 针对功能全链条难以追溯的问题,主要原因是EA工具的局限性,不能完美适配面向SOA的整车E/E架构开发,同样可以通过开发插件的方式进行弥补,将EA上的模型元素建立关联关系后,选择相应的功能或某个特定的UC,实现从Function-UC-PC-Module-SWC的全链条导出,如图17所示。

(4) 由于在开发过程中使用到了EA以及PREvision两套工具,其相互独立,未形成联动,导致创建SWC和SWC的接口等工作在两套工具上都进行了一遍,重复工作在开发过程中费时费力,且容易出现差

错,一旦其中一种工具上进行了修改,很难同步到另一种工具上,导致后期围护工作量加大。为解决此问题,可以对EA上的SWC模型进行导出标准格式的Excel文件,同时在PREvision上也进行二次开发,将Excel文件导入到PREvision上进行自动建模,保证工具链之间建模内容的一致性,避免重复的操作,提升工作效率。通过EA二次开发插件所导出Excel格式文件如图18所示。

FO	Function ID	Function	Use case	PC used	SWC	Related Module
1				Provide View for LAC Disable/Enable Status	ADAS_VDU_JACCControl	AXXA Demo Module
2				Provide LAC setting	ADAS_VDU_JACCControl	AXXA Demo Module
3				LAC Control	ADAS_VDU_JACCControl	AXXA Demo Module
4				Handle LAC set Request from button	ADAS_VDU_CameraControl	AXXA Demo Module
5				Handle LAC set Request from button	ADAS_VDU_CameraControl	AXXA Demo Module
6				Provide Driver information view in Instrument Screen	ADAS_VDU_JACCControl	AXXA Demo Module
7				Provide Driver information view in Instrument Screen	ADAS_VDU_JACCControl	AXXA Demo Module
8				Provide LAC setting	ADAS_VDU_JACCControl	AXXA Demo Module
9				Provide View for LAC Disable/Enable Status	ADAS_VDU_JACCControl	AXXA Demo Module
10				LAC Control	ADAS_VDU_CameraControl	AXXA Demo Module
11				Handle LAC set Request from button	ADAS_VDU_CameraControl	AXXA Demo Module
12				Handle LAC set Request from button	ADAS_VDU_CameraControl	AXXA Demo Module
13				Provide Driver information view in Instrument Screen	ADAS_VDU_JACCControl	AXXA Demo Module
14				Provide Driver information view in Instrument Screen	ADAS_VDU_JACCControl	AXXA Demo Module
15				No intervention during forward driving		
16				No intervention during forward driving		

图17 EA二次开发—模型元素全链条导出示例

Service Name 服务接口名称	Service Interface Element Name 服务接口元素名称	Service Interface Element Type 服务接口元素类型	Operation Argument Name 方法参数名称	Operation Argument Direction 方法参数发送方向	Send Strategy 发送策略	Epsilon Value(阈值) 参数(阈值)	Data Type Reference 参考数据类型
BO_SA_FDD_AccCharg	Dt_ChargeStatus	Event					BO_ChargePortStatus_er
	Dt_NotifyCycleAccChargeEvent	Event			cyclic on change	50	BO_ChargePortStatus_er
	Dt_GetAccChargeInfo	F&F Method		IN	cyclic on change	50	BO_ChargePortStatus_er
	Dt_FRACChargePortOpen	F&F Method		IN			BO_ChargePortStatus_er
BO_SA_FDD_AccCharg	Dt_AccChargeStatus	Event					BO_ChargePortStatus_er
	Dt_NotifyOnChgAccChargeEvent	Event			update on chan		BO_ChargePortStatus_er
	Dt_GetAccChargeInfo	F&F Method		IN			BO_ChargePortStatus_er
	Dt_FRACChargePortOpen	F&F Method		IN			BO_ChargePortStatus_er

图18 EA二次开发插件导出标准格式Excel实现示例

4 结束语

在面向SOA架构开发的过程中,EA和PREvision是2款可以提高开发及管理效率的数智化转型工具。利用EA工具可以有效开展功能架构设计工作,通过UML语言可很好的阐述功能定义和功能实现逻辑。PREvision可以很好的对SWC进行设计并导出ARXML供下游进行软件开发,并提供优秀的追溯性。同时结合两款软件的不足之处和局限性,有针对性的进行二次开发,解决了开发流程中所存在的问题,打通了工具之间的瓶颈,使其更加适配于项目本身,提升设计效率,改善设计质量。

参考文献

[1] 李丹,吕颖,李骏,等.面向服务的体系架构[J].汽车文摘,2021(10):52-57.
 [2] 王东,李井波,张晨晖,等.一种基于SOA架构的总线系统:CN112822080B[P].2022-09-16.
 [3] 编辑部.德国大众旗下CARIAD携手意法半导体面向软件定义汽车[J].单片机与嵌入式系统应用,2022,22(10):2-2.

[4] 金华旺,张桂新.工业企业数智化转型关键问题及推进路径研究[J].信息系统工程,2022,346(10):14-17.
 [5] KUGELE S, HETTLER D, PETER J.Data-centric communication and containerization for future automotive software architectures[C]. IEEE International Conference on Software Architecture (ICSA), 2018.
 [6] 李升凯,吴长水.智能驾驶硬件计算平台及CAN通信软件设计[J].软件工程,2022,25(7):51-54.
 [7] 唱佳韵,孙明阳.汽车文化APP界面设计研究——以红旗汽车文化APP界面设计为例[J].工业设计,2022,195(10):64-66.
 [8] 詹克旭.基于AUTOSAR架构的汽车电子软件产品的开发方法[J].汽车电器,2021(11):73-76.
 [9] 王飞飞,刘梦梦,翟晓兰.智能网联汽车测试示范区发展模式探究[J].汽车实用技术,2022(13):26-30.
 [10] 李震,刘敏.基于AUTOSAR的整车电子电气架构设计方法[J].机电一体化,2012(11):73-76.
 [11] 伏军锋.汽车电气系统设计[J].汽车电器,2010(1):1-5.
 [12] 陈敏.基于EA的电子电气架构功能设计探讨[J].汽车电器,2022(6):38-41+45.
 [13] FENG J T. UML-Based Radar Software Design[C]//Proceedings of 5th International Conference on Electrical & Electronics Engineering and Computer Science(ICEECS 2018), 2018:426-429.
 [14] 王永辉.基于PREvision的汽车电子电气架构设计介绍[J].汽车实用技术,2019,294(15):111-112.
 [15] 匡小军,唐香蕉,周涛,等.基于PREvision的汽车电子电气架构工具链研究[J].汽车电器,2019,372(8):62-64.

(责任编辑 慧慧)

【作者简介】

金劭南(1995-),男,中国第一汽车股份有限公司研发总院,硕士,研究方向为整车电子电气架构设计。
 高仕宁(1984-),男,中国第一汽车股份有限公司研发总院,硕士,研究方向为整车电子电气架构设计。
 李超(1986-),男,中国第一汽车股份有限公司研发总院,硕士,研究方向为整车电子电气架构设计。
 吴振举(1982-),男,中国第一汽车股份有限公司研发总院,学士,研究方向为整车电子电气架构设计。
 陈泓宇(1990-),男,中国第一汽车股份有限公司研发总院,硕士,研究方向为整车电子电气架构设计。
 柳菁(1989-),女,中国第一汽车股份有限公司研发总院,学士,研究方向为整车电子电气架构设计。